

## REMARKS

Claims 1-18 are pending in the application. In the final Office Action of April 20, 2006, the Examiner made the following disposition:

- A.) Rejected claims 1-6 and 8-16 under 35 U.S.C. §103(a) as being allegedly unpatentable over *Kanai* (U.S. Patent No. 5,339,410) (“*Kanai*”) in view of *Bhansali, et al.* (U.S. Patent No. 6,006,239) (“*Bhansali*”).
- B.) Rejected claims 7, 17, and 18 under 35 U.S.C. §103(a) as allegedly being unpatentable over *Sorge et al.* (U.S. Patent No. 6,691,281) (“*Sorge*”) in view of *Zellweger et al.* (U.S. Patent No. 6,185,582) (“*Zellweger*”) and further in view of *Bhansali, et al.*

Applicant respectfully traverses the rejection and addresses the Examiner’s disposition below.

- A.) Rejection of claims 1-6 and 8-16 under 35 U.S.C. §103(a) as being allegedly unpatentable over *Kanai* (U.S. Patent No. 5,339,410) (“*Kanai*”) in view of *Bhansali, et al.* (U.S. Patent No. 6,006,239) (“*Bhansali*”):

Applicants respectfully disagree with the rejection.

Claims 1, 8, and 11 each claim subject matter relating to overriding without deleting the original content of a first cell with a first value and then recalculating the cells. Then, the original content of a second cell is overridden, without being deleted, with a second value and the cells are recalculated again. Then, the original content of the first cell is automatically restored, while the second value is maintained in the second cell.

The first cell keeps the original content of the first cell in the first cell while the original content of the first cell is overridden, and the second cell keeps the original content of the second cell in the second cell while the original content of the second cell is overridden. Thus, the original content of the first cell is automatically restored independently of restoring the original content of the second cell, and after multiple recalculations.

This is clearly unlike *Kanai* in view of *Bhansali*, which fails to disclose or suggest overriding without deleting an original content of a cell, and then automatically restoring the original content to the cell after multiple recalculations, and independently of restoring an overridden, original content to a second cell. *Kanai* teaches a method of bidirectional recalculation of cells, that is recalculations can be performed in both directions. For example, the value in cell A may automatically change when a user inputs a new value into cell B, and the value in cell B may automatically change when a user inputs a new value into cell A.

Contrary to the Examiner’s argument, *Kanai* does not override, without deleting, the

original content of a cell. As *Kanai* clearly describes with reference to the example in Figures 2 and 3, when the value in cell X2 is changed, the value in cell Y automatically changes. This is because the value in cell Y is calculated based on the formula: cell Y's value = cell X1's value + cell X2's value. *Kanai* Figures 2 and 3; 3:60-4:24. Similarly, as shown in Figure 4, when the value in cell Y changes, the value in cell X1 automatically changes. This is because the value in cell X1 depends on the value of cell Y. *Kanai* Figure 4.

The Examiner argues that these same Figures and text passages from *Kanai* teach overriding, without deleting, the original content of a cell. However, nowhere do these passages make such a teaching. *Kanai*'s cells X1, X2, and Y are different cells. Their values are influenced by the values of other cells (e.g.,  $Y = X1 + X2$ ). Therefore, their values are automatically recalculated when the value in an influencing cell changes. It appears that the Examiner has misinterpreted the term "recalculate" to mean override. However, recalculate does not mean override as used in Applicant's claims. Nowhere does *Kanai* suggest that its cell values are overridden without being deleted. Instead, *Kanai* merely teaches automatically recalculating cells.

*Bhansali* also fails to disclose or suggest overriding without deleting an original content of a cell, and then automatically restoring the original content to the cell after multiple recalculations, and independently of restoring an overridden, original content to a second cell.

*Bhansali* allows multiple users to edit a spreadsheet document by maintaining a local copy of the spreadsheet at each user's computer. A central copy is maintained on a centralized storage disk. Each user's computer includes a "memory change log" that keeps track of changes made by that user. When a user saves the user's changes, that user's memory change log is appended to a "disk change log" at the central storage disk. The changes of the various users are made to the centralized copy of the spreadsheet by sequentially processing each of the memory change logs that have been appended to the disk change log. When two users' edits conflict, either a user chooses the winning edit or the most recent edit is used. When a user's edit loses, an inverse of the edit is performed such that the end result is effectively that the edit never took place (e.g., a losing edit that inserted a row is inverted by subsequently deleting the row).

In *Bhansali*, a record of losing edits is maintained at a user's computer in a "disk undo log" (losing edits are undone in the user's local copy). When the user saves the spreadsheet, the user's disk undo log is also appended to the centralized disk change log. Thus, when the centralized copy of the spreadsheet is updated, "memory change log" edits are implemented as well as "disk undo log" edits to correct any conflicts by implementing inverse actions of the

losing edits. *Bhansali* 11:45-67.

Thus, unlike Applicant's claimed invention, *Bhansali* fails to disclose or suggest overriding without deleting the original content of a cell with a user inputted value, wherein the cell keeps the original content in the cell. Instead, *Bhansali* allows a user to overwrite and delete the original value of a cell in the user's local copy. The original content of the cell can be replaced by performing an inverse action. To perform the inverse action, *Bhansali* looks to its "disk undo log" to identify the losing action and then implements the losing action's inverse. Thus, the original content of a cell is not maintained in the cell, but in the disk undo log. *Bhansali* deletes the original content of a cell and then restores the original content with information that it retrieves from its disk undo log. Accordingly, *Bhansali* also fails to disclose or suggest overriding without deleting the original content of a cell with a user inputted value, wherein the cell keeps the original content in the cell.

Therefore, *Kanai* in view of *Bhansali* still fails to disclose or suggest claims 1, 8, and 11.

Claims 2-6, 9-10 and 12-16 depend directly or indirectly from claim 1, 8, or 11 and are therefore allowable for at least the same reasons that claims 1, 8, and 11 are allowable.

B.) Rejection of claims 7, 17, and 18 under 35 U.S.C. §103(a) as allegedly being unpatentable over *Sorge et al.* (U.S. Patent No. 6,691,281) ("*Sorge*") in view of *Zellweger et al.* (U.S. Patent No. 6,185,582) ("*Zellweger*") and further in view of *Bhansali, et al.*:

Applicant respectfully disagrees with the rejection.

Regarding claims 7 and 17:

Independent claims 7 and 17 each claim a plurality of cells each comprising a formula and a last result. A plurality of values are received for the plurality of cells. The values are stored in the last result of the plurality of the cells such that the values are used during recalculation instead of the formulas and such that each of the formulas for the plurality of the cells can be restored independently of other of the plurality of cells.

This is clearly unlike *Sorge* in view of *Zellweger* and further in view of *Bhansali*, which fails to disclose or suggest a cell comprising a formula and a last result, wherein the value stored in the last result is used to recalculate the cell instead of the formula. In fact, nowhere do the cited references, taken singly or in combination, teach using a value stored in a last result to recalculate a cell instead of the cell's formula. The Examiner cites several passages from *Sorge*, alleging that *Sorge* teaches using a value stored in the last result portion of a cell instead of the

cell's formula. However, none of the cited passages even relate to this. (*Office Action of 4/20/2006*, pages 8-9). Instead, the cited passages merely describe that the cells of a spreadsheet-version of a spreadsheet document can be replaced (i.e., overwritten) with the cells of an HTML-version of the document (*Sorge* 4:10-20) and describe that the spreadsheet-version of the document can be conveniently converted to the HTML version (*Sorge* 10:45-57). Nowhere do these or other passages from *Sorge* discuss using a last result portion of a cell to recalculate instead of the cell's formula.

As discussed above, *Sorge* teaches maintaining two versions of a spreadsheet document (a spreadsheet version and an HTML version) and replacing the cells in one version with the cells in the other version. *Sorge*, 4:10-20. When a value and formula are inserted into a spreadsheet cell from the HTML document, the value and formula overwrite the original content of the spreadsheet cell. When the cell is recalculated, *Sorge* uses the cell's formula.

Unlike claims 7 and 17, nowhere does *Sorge* discuss using a received, stored value of a cell to recalculate the cell instead of a formula of the cell. Instead, *Sorge* merely teaches replacing the original content of a cell with a value and a formula from another document. Further, *Sorge* merely teaches that a cell's formula is used when recalculating.

*Zellweger* fails to disclose or suggest using a cell's stored value to recalculate the cell instead of the cell's formula. *Bhansali* also fails to discuss recalculating a cell using a received, stored value instead of the cell's formula.

Therefore, *Sorge* in view of *Zellweger* and further in view of *Bhansali* fails to disclose or suggest claims 7 and 17.

Regarding claim 18:

Independent claim 18 claims a cell having a first storage area that stores a formula and a second storage area that stores a numerical value that temporarily overrides a formula so that the numerical value is used instead of the formula during recalculation.

As discussed above, *Sorge*, *Zellweger*, and *Bhansali*, taken singly or in combination, fail to disclose or suggest temporarily overriding the content of a cell and fails to disclose or suggest recalculating a cell using a value instead of the cell's formula. Instead, *Sorge* merely teaches overwriting a spreadsheet cell's contents, including its formula, with a value and formula from an HTML document. When *Sorge*'s spreadsheet cell's original contents are overwritten, they are not temporarily overridden -- in fact, they no longer exist.

Therefore, *Sorge* in view of *Zellweger* and further in view of *Bhansali* fails to disclose or suggest claim 18.

## CONCLUSION

In view of the foregoing, it is submitted that claims 1-18 are patentable. It is therefore submitted that the application is in condition for allowance. Notice to that effect is respectfully requested.

Respectfully submitted,

/Christopher P. Rauch/ (Reg. No. 45,034)  
Christopher P. Rauch  
SONNENSCHN, NATH & ROSENTHAL LLP  
P.O. Box #061080  
Wacker Drive Station - Sears Tower  
Chicago, IL 60606-1080  
Telephone 312/876-2606  
Customer #26263  
Attorneys for Applicant(s)